

BASES DE DATOS 2

Ingenieria Informatica-Cuarto Curso

Profesor: Luis A. Glez. Ares

 (Fichero sin acentos para facilitar su lectura
 en distintas plataformas)

EJEMPLOS DE SQL*Loader

 Estos varios ejemplos de utilizacion del SQL*Loader, con alternativas
 suficientes para diversos casos de carga, incluyendo la utilización de
 secuencias.

Hay otras opciones y posibilidades, pero habra que consultar el manual
 para su correcta utilizacion (Utilities).

Creamos una tabla de prueba:

```
SQL> CREATE TABLE p1
      ( cd      CHAR(3),
        nme     CHAR(10),
        sal     NUMBER,
        dia     DATE);
```

cuya descripcion es:

```
SQL> desc p1
Name                               Null?    Type
-----
CD                                  CHAR(3)
NME                                 CHAR(10)
SAL                                 NUMBER
DIA                                 DATE
```

Ejemplo 1

 - Los datos que queremos introducir estan en el fichero p1.dat
 que tiene solo las cuatro filas que aparecen bajo los guiones:

```
123456789012345678901234
```

```
-----
101 fila 1      50 980423
102 fila 2      51 980424
103              52 980425
104 fila 4      980426
```

- El fichero de control que usamos es p1.ctl que es el siguiente:

```
load data
infile 'p1.dat'
into table p1
(cd position(1:3) char,
 nme position(5:14) char,
 sal position(15:17) integer external,
 dia position(19:24) date 'YYMMDD')
```

- Ejecutamos el SQL*Loader:

```
$ sqlldr [usuario] control=p1
```

- Siempre comprobaremos el fichero de log (aqui: p1.log).

- Realizamos una llamada desde el operativo a sqlplus:

```
$ sqlplus [usuario] @p1
```

- Con esto ejecutamos el fichero p1.sql que contiene:

```
SELECT *
FROM   p1;
exit
```

- Y obtenemos:

Conectado a:

Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - 64bit Production

CD	NME	SAL	DIA
101	fila 1	50	23/04/98
102	fila 2	51	24/04/98
103		52	25/04/98
104	fila 4		26/04/98

4 filas seleccionadas.

Desconectado de Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - 64bit Production

- Luego podemos comprobar los valores nulos:

```
SQL> SELECT *
      FROM   p1
      WHERE  nme IS NULL;
```

CD	NME	SAL	DIA
103		52	25/04/98

```
SQL> SELECT *
      FROM   p1
      WHERE  sal IS NULL;
```

CD	NME	SAL	DIA
104	fila 4		26/04/98

Ejemplo 2

- Ahora usamos un fichero de control que contiene las ordenes de la carga y los datos a introducir, que estan delimitados por comas.
El fichero de control funcionara sobre una tabla vacia, pero P1 realmente ya tiene filas, lo que originara un error.

- El fichero de control es p11.ctl:

```
load data
infile *
into table p1
fields terminated by ',' optionally enclosed by ''
```

```
(cd, nme, sal, dia date 'YYMMDD')
begindata
200, filauno, 3,980427
201,"fila's", 4,980428
```

- Ejecutamos el SQL*Loader:

```
$ sqlldr [usuario] control=p11
```

```
SQL*Loader: Release 11.1.0.6.0
```

```
SQL*Loader-601: Para la opción INSERT, la tabla debe estar vacía. Error en la tabla P1
```

- Nos da un error debido a que la tabla no esta vacia. Entramos en sqlplus y eliminamos todas las filas de P1:

```
SQL> delete from p1;
```

Si deseamos incorporar nuevas filas a una tabla que tiene algunas, debemos usar APPEND despues de INFILE (se vera mas abajo).

- Salimos y ejecutamos de nuevo el SQL*Loader con el fichero de control p11:

```
$ sqlldr [usuario] control=p11
```

```
SQL*Loader: Release 11.1.0.6.0
```

```
Punto de validación alcanzado - recuento de registros lógicos 2
```

- Comprobamos el contenido de la tabla:

```
$ sqlplus [usuario] @p1
```

Conectado a:

```
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - 64bit Production
```

```
CD  NME          SAL DIA
---  -
200 filauno      3 27/04/98
201 fila's      4 28/04/98
```

```
2 filas seleccionadas.
```

```
Desconectado de Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - 64bit Production
```

- Ahora decidimos introducir mas filas en la tabla p1. Para ello usamos el fichero de control p12.ctl:

```
load data
infile *
append                                <----- APPEND
into table p1
fields terminated by ','
(cd, nme, sal, dia date 'YYMMDD')
begindata
400, masfilas, 3,980427
401, masfilas, 4,980428
```

- Ejecutamos el SQL*Loader:

```
$ sqlldr [usuario] control=p12
```

```
SQL*Loader: Release 11.1.0.6.0
```

Punto de validación alcanzado - recuento de registros lógicos 2

- Comprobamos el contenido de la tabla:

```
$ sqlplus [usuario] @p1
```

Conectado a:

Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - 64bit Production

CD	NME	SAL	DIA
200	filauno	3	27/04/98
201	fila's	4	28/04/98
400	masfilas	3	27/04/98
401	masfilas	4	28/04/98

4 filas seleccionadas.

Desconectado de Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - 64bit Production

Ejemplo 3

Ejemplo de SQL Loader con uso de secuencia para la carga de una columna.

- Creamos la tabla dp:

```
CREATE TABLE dp
( deptno NUMBER(2),
  dname VARCHAR2(14),
  loc VARCHAR2(12));
```

- Creamos una secuencia de nombre sqdp:

```
SQL> CREATE SEQUENCE sqdp INCREMENT BY 2 START WITH 10;
```

- Creamos un fichero de control llamado dp1.ctl:

```
$ cat dp1.ctl
load data
infile *
into table dp
fields terminated by ',' optionally enclosed by '"'
TRAILING NULLCOLS
(dname, loc, deptno "sqdp.NEXTVAL")
begindata
CONTABILIDADE, VIGO
TURISMO, LUGO
```

- Ejecutamos SQL*Loader con el fichero de control dp1.ctl:

```
$ sqlldr [usuario] control=dp1
```

```
SQL*Loader: Release 11.1.0.6.0
```

Punto de validación alcanzado - recuento de registros lógicos 2

- Comprobamos los datos dela tabla dp:

```
SQL> SELECT * FROM dp;
```

DEPTNO	DNAME	LOC
10	CONTABILIDADE	VIGO
12	TURISMO	LUGO

2 filas seleccionadas.

- Ahora usamos un fichero de control con los datos posicionales y suponiendo que la tabla pueda tener filas:

```
$ cat dp2.ctl
load data
infile *
append
into table dp
(deptno "sqdp.NEXTVAL",
 dname position(1:14),
 loc position(15:26))
begindata
VENTAS      A CORUÑA
MARKETING   OURENSE
```

- Ejecutamos SQL*Loader con el fichero de control dp2.ctl:

```
$ sqlldr [usuario] control=dp2
```

```
SQL*Loader: Release 11.1.0.6.0
```

Punto de validación alcanzado - recuento de registros lógicos 2

- Comprobamos los datos dela tabla dp:

```
SQL> SELECT * FROM dp;
```

DEPTNO	DNAME	LOC
14	VENTAS	A CORUÑA
16	MARKETING	OURENSE
10	CONTABILIDADE	VIGO
12	TURISMO	LUGO

4 filas seleccionadas.

Ejemplo 4

Cargando datos decimales.

Hay que tener cuidado con las variables de entorno de Oracle que determinan el lenguaje local (NLS_LANG).

Creamos una tabla con una columna decimal:

```
SQL> CREATE TABLE d2p
      ( c1 NUMERIC(7,2),
        c2 CHAR(5))
```

```
INSERT INTO d2p VALUES (123.2, 'dd');
```

```
SQL> SELECT * FROM d2p;
```

```
      C1 C2
-----
    123,2 dd
```

1 fila seleccionada.

La visualizacion de los decimales es con una coma, aunque el INSERT permite usar el punto como separador decimal.

Creamos el fichero de control d2p.ctl

```
-- Para introducir valores decimales con NLS_LANG=SPANISH (y variantes)
LOAD DATA
INFILE *
APPEND
INTO TABLE d2p
  FIELDS TERMINATED BY ":" OPTIONALLY ENCLOSED BY '"'
(c1, c2)
BEGINDATA
1,1:1
2,5:2
2,4:3
21,2:un
22,22:dous
1500,5:tres
```

Ejecutamos SQL*Loader:

```
sqlldr [usuario] control=d2p.ctl
```

```
SQL*Loader: Release 11.1.0.6.0
```

Punto de confirmación alcanzado - recuento de registros lógicos 6

```
SQL> SELECT * FROM d2p;
```

```
      C1 C2
-----
    123,2 dd
      1,1 1
      2,5 2
      2,4 3
     21,2 un
     22,22 dous
    1500,5 tres
```

7 filas seleccionadas.